Design of an Online Testable Ternary Circuit from the Truth Table

N. M. Nayeem and J. E. Rice

Dept. of Math & Computer Science
University of Lethbridge, Lethbridge, Canada
{noor.nayeem,j.rice}@uleth.ca

Abstract. This paper presents a new approach for converting a ternary reversible circuit implemented from a truth table into an online testable circuit. Our approach adds three extra lines to the given circuit, inserts Feynman gates and M-S gates, and replaces the ternary Toffoli gates (KP-m gates) with TKP-(m+1) gates. Our approach works with only 2×2 gates and 1×1 gates and covers a higher number of detectable faults. Preliminary work shows fault coverage of 84.89% when the approach is applied to a testable ternary half adder.

Keywords: Reversible logic, ternary circuits, online testing

1 Introduction

Circuits built using traditional logic lose information during computation, which is dissipated as heat [1]. One solution to this loss of information is reversible logic. In particular, Bennett showed that a circuit consisting of only reversible gates dissipates zero energy [2]. There has, however, been little work on testability of reversible circuits, and even less in the area of multiple-valued testable reversible circuits. There is motivation to develop research in multiple-valued reversible logic, as this can provide a stepping stone to up-and-coming quantum technologies since quantum computing is inherently multiple-valued [3]. Previous work in online testing for reversible circuits includes our own in [4] as well as work in the Boolean domain such as [5] and [6].

In this work we introduce a new approach for converting a ternary reversible circuit into an online testable circuit, with fault coverage improvements over previous work reported in [4]. We emphasize that this work is ongoing, and preliminary results are reported here.

2 Background

2.1 Online Testing

Those of us in the field of computing are aware that testing is required to ensure quality and reliability. This applies to reversible logic circuits as well as to traditional logic circuits. According to [7], testing can be performed in the following ways: online; that is, while the circuit is operating normally; offline;

or during a period while the circuit is not in use; or using some combination of both online and offline testing. The work proposed here is for an online testing approach, thus we would not require the circuit to be taken out of operation for the fault-detection to take place.

2.2 Fault Models

There are several fault models in reversible logic some of which include the missing, repeated and reduced gate fault models [8]. An additional and technology-independent fault model referred to as the bit fault model is used in various works including [5] and [6]. In this model a fault, possibly in a gate, would change the behavior of the gate's outputs. A single-bit fault is reflected on exactly one output of a gate, changing the correct value of the output to a faulty value. This model is somewhat reminiscent of the stuck-at fault model. We use this single-bit model in this work, although we note that the use of the term "bit" is not entirely accurate for ternary logic. The original concept of this model is still valid, however, as we are identifying the situation when a fault is reflected on exactly one output of a gate.

2.3 Ternary Galois Field Logic

The Ternary Galois Field (TGF) consists of {0, 1, 2} and two operations, addition modulo 3 and multiplication modulo 3. We denote addition modulo 3 by $\oplus$ and multiplication modulo 3 by the absence of any operator. For a ternary variable a, we have $a = a \oplus 3$ and $aaa = a$. According to [9], a ternary variable a has six basic literals: a, $a^{+1} = a \oplus 1$, $a^{+2} = a \oplus 2$, $a^{12} = 2a$, $a^{01} = 2a \oplus 1$, and $a^{02} = 2a \oplus 2$.

2.4 Reversible Ternary Gates

We define here the ternary reversible gates which are required for this paper. A 1-qutrit permutative gate [10] is defined as $\{a\} \rightarrow \{b = a^Z\}$ where $Z \in \{+1, +2, 12, 01, 02\}$ as shown in Fig. 1(a). For example, if $Z = +1$, then $b = a^{+1} = a \oplus 1$.

The Feynman gate has the input vector $[a_1, a_2]$ and output vector $[b_1 = a_1, b_2 = a_1 \oplus a_2]$. The modified Feynman gate discussed in [11] is very similar to the Feynman gate with the exception that $b_2 = 2a_1 \oplus a_2$. The Feynman gate and its modified version are shown in Fig. 1(b) and Fig. 1(c).

A 2-qutrit Muthukrishnan-Stroud (M-S) gate [10] is defined as mapping the input vector $[a_1, a_2]$ to the output vector $[b_1 = a_1, b_2 = r]$ where $r = a_2^Z$ if $a_1 = 2$; otherwise $r = a_2$. Here a_1 is the controlling input and a_2 is the controlled input. An M-S gate is shown in Fig. 1(d).

Khan and Perkowski proposed a ternary Toffoli gate [10]; however its behaviour is somewhat different from the commonly accepted behaviour for a Toffoli gate and so to avoid confusion we refer to this gate as a KP gate. A $(p + 1)$-qutrit KP gate is shown in Fig. 1(e). This gate maps the input vector $[a_1, a_2, \ldots, a_{p+1}]$ to the output vector $[b_1 = a_1, b_2 = a_2, \ldots, b_p = a_p, b_{p+1} = r]$

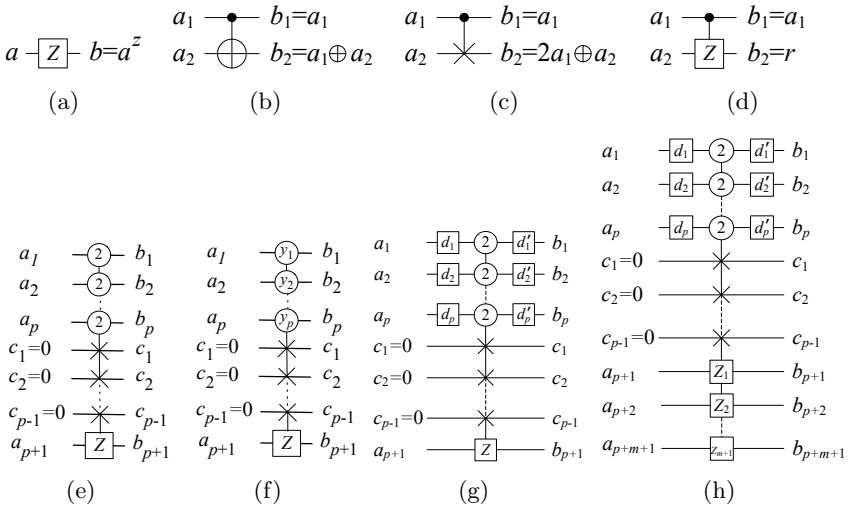

Fig. 1: (a) A permutative gate, (b) Feynman gate, (c) modified Feynman gate, (d) M-S gate, (e) $(p+1)$-qutrit KP gate, (f) $(p+1)$-qutrit generalized KP gate, (g) equivalent representation of a generalized KP gate, and (h) KP-m gate.

where $r = a_{p+1}^{Z}$ if $a_1 = a_2 = \ldots = a_p = 2$; otherwise $r = a_{p+1}$. Here $a_1, a_2, \ldots, a_p$ are controlling inputs and a_{p+1} is the controlled input.

Khan and Perkowski also proposed a generalized KP gate [10] as shown in Fig. 1(f), which is very similar to the KP gate. In this gate $r = a_{p+1}^{Z}$ if $a_1 = y_1, a_2 = y_2, \ldots, a_p = y_p$; otherwise $r = a_{p+1}$. An equivalent representation of this gate is shown in Fig. 1(g). This version is built using the KP gate and 1-qutrit permutative gates where $d_j = 2 - y_j$ and $d_j + d'_j = 0$ for $j = 1, 2, \ldots, p$. The permutative gate $+d_j$ is used to change the controlling input of KP gate to 2, and the permutative gate $+d'_j$ restores the controlling value.

We can extend the KP gate and the generalized KP gate for multiple controlled inputs. For example, a $(p+1)$-qutrit generalized KP gate with $m+1$ controlled inputs, denoted as a KP-m gate, is shown in Fig. 1(h). The KP-m gate has the input vector $[a_1, a_2, \ldots, a_p, a_{p+1}, \ldots, k_{p+m+1}]$ and the output vector $[b_1 = a_1, b_2 = a_2, \ldots, b_p = a_p, b_{p+1} = r_1, b_{p+2} = r_2, \ldots, b_{p+m+1} = r_{m+1}]$ where $r_k = a_{p+k}^{Z_k}$ if $a_1 = y_1$, $a_2 = y_2$, ..., a_p=y_p and $Z_k \in \{+1, +2, 12, 01, 02\}$ for $k = 1, 2, \ldots, m+1$; otherwise $r_k = a_{p+k}$. Here, $a_1, a_2, \ldots, a_p$ are controlling inputs and $a_{p+1}, a_{p+2}, \ldots, a_{p+m+1}$ are controlled inputs. Like the $(p+1)$-qutrit KP gate and generalized KP gate, a $(p+1)$-qutrit KP-m gate also requires $p-1$ constant lines.

In order to design the testable ternary circuit we add the following constraints to the KP-m gate: $Z_k \in \{+1, +2\}$ $(k = 1, 2, \ldots, m)$ and $Z_{m+1} = Z_1 \oplus Z_2 \oplus \ldots \oplus Z_m$. To distinguish this gate from the KP-m gate we call this gate a TKP-m gate (testable KP-m). The symbol of a $(p+1)$-qutrit TKP-m gate is shown in Fig. 2 (on the following page).

2.5 Synthesis of Ternary Reversible Circuits

Several approaches such as [12], [13] and [14] have been proposed for synthesis of ternary reversible circuits. In this section we briefly describe an approach [10] to realize a ternary circuit from the truth table.

Consider a ternary function with p input variables $x_1, x_2, \ldots, x_p$ and q output variables $f_1, f_2, \ldots, f_q$. An empty cascade with p input lines $(I_1, I_2, \ldots, I_p)$, $p-1$ constant lines $(I_{p+1}, I_{p+2}, \ldots, I_{2p-1})$ and q output lines $(I_{2p}, I_{2p+1}, \ldots, I_{2p+q-1})$ is generated. For each input combination $x_1 x_2 \ldots x_p$ ($x_i \in \{0, 1, 2\}$, for $i = 1, 2, \ldots, p$) with $m+1$ outputs ($0 \leq m < q$) having values 1 or 2 in the truth table, a $(p+1)$-qutrit generalized KP-m gate with $x_1 = y_1, x_1 = y_1, \ldots, x_p = y_p$ is added to the circuit. Each controlling input a_i of the gate is connected to the input line I_i. For each $f_j = 1$ (or 2) $j = 1, 2, \ldots, q$, a controlled input is connected to the output line I_{2p+j-1} with $Z = +1$ (or $+2$). Although we have described this approach using generalized KP-m gates the circuit can also be generated using KP-m gates and permutative gates.

As an example, given the truth table of a ternary function with two input variables (x_1 and x_2) and two output variables (f_1 and f_2) as shown in Table 1, a ternary circuit is implemented as shown in Fig. 3. For the first input combination, a 3-qutrit KP-1 gate is added. The controlling inputs of this gate are connected to lines I_1 and I_2 and the controlled inputs are connected to lines I_4 and I_5. Two 3-qutrit KP gates are added for the second and third input combinations. No more gates are added since both f_1 and f_2 have values 0 for all other input combinations.

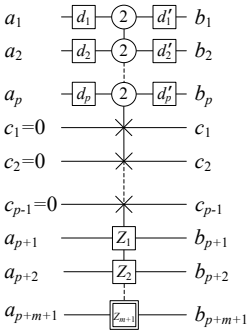

Fig. 2: $(p+1)$-qutrit TKP-m gate.

x_1	x_2	f_1	f_2
0	0	1	1
0	1	1	0
0	2	0	2
1	0	0	0
1	1	0	0
1	2	0	0
2	0	0	0
2	1	0	0
2	2	0	0

Table 1: Truth table of a ternary function.

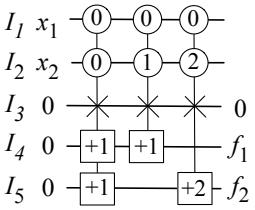

Fig. 3: A ternary circuit.

3 Our Approach

3.1 Design

Consider a reversible ternary circuit generated from the truth table as discussed in Section 2.5. A circuit generated in this way consists of only permutative gates, KP-m gates, and generalized KP-m gates. If the circuit has p input lines and q

output lines, then the circuit also has $p-1$ constant lines. We refer to the input lines as $I_1, I_2, \ldots, I_p$, the constant lines as $I_{p+1}, I_{p+2}, \ldots, I_{2p-1}$, and the output lines as $I_{2p}, I_{2p+1}, \ldots, I_{2p+q-1}$. If the initial values and final values of any input line (or constant line) are not the same, then the permutative gates and M-S gates are added to restore the initial value at the end of the corresponding line. The following approach converts such circuit into an online testable circuit.

Our proposed approach requires three extra lines, L_1, L_2, and L_3, each of which is initialized with a zero. All permutative gates found in the given circuit are retained. Each KP-m gate is replaced by a TKP-$(m+1)$ gate. The connections of the TKP-$(m+1)$ gate remain the same as that of KP-m gate with the last controlled input connected to L_1.

For each input line in the given circuit, this approach adds a Feynman gate and a modified Feynman gate at the beginning and at the end of the circuit, respectively. For each such gate, the controlling input is connected to I_u (for $u = 1, 2, \ldots, p$) and controlled input is connected to L_2. At the end of each constant line we first add a permutative ($Z = +2$) gate, then an M-S gate ($Z = +1$) with controlling input connected to I_v (for $v = p+1, p+2, \ldots, 2p-1$) and controlled input connected to L_3. Another permutative gate ($Z = +1$) is also added on the constant line to restore the value. Finally, at the end of each output line a modified Feynman gate is added with controlling input connected to I_w (for $w = 2p, 2p+1, \ldots, 2p+q-1$) and controlled input connected to L_1. This approach requires the addition of p Feynman gates, $p+q$ modified Feynman gates, $p-1$ M-S gates and $2p-2$ permutative gates.

If a single fault occurs in any gate other than the Feynman gate and the modified Feynman gate, then L_1 or L_2 will be non-zero, or L_3 will not be equal to $1 \oplus 2 \oplus \ldots \oplus p-1$. If no fault occurs in the circuit, then L_1 and L_2 will remain 0, and L_3 will be equal to $1 \oplus 2 \oplus \ldots \oplus p-1$. Thus existence of a fault can be detected by examining the values of L_1, L_2 and L_3.

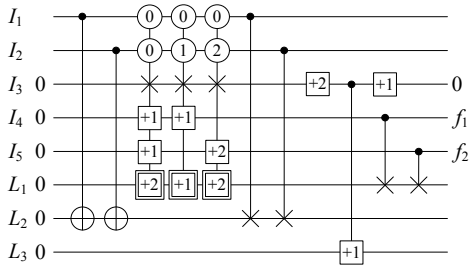

Fig. 4: An online testable ternary circuit.

The following example describes this approach. For a ternary circuit as shown in Fig. 3, our proposed approach generates an online testable circuit as shown in Fig. 4. In the testable circuit three extra lines are added and the KP-m gates are replaced by TKP-$(m+1)$ gates. In addition, two Feynman gates, four modified Feynman gates, one M-S gate and two permutative gates are added.

3.2 Fault Detection

Our proposed approach makes use of the TKP-$(m+1)$ gates and generalized TKP-$(m+1)$ gates, which can be decomposed into M-S gates and permutative gates. In this section, we consider a low level design of our testable circuit consisting of 1×1 gates and 2×2 gates which are permutative gates, M-S gates, Feynman gates, and the modified Feynman gates.

A single fault on a line can propagate to several lines via M-S gates. This causes multiple faults in the circuit. Consider an M-S gate as shown in Fig. 1(d). A fault in the controlling input a_1 (or controlled input a_2) of an M-S gate affects this gate since it causes b_1 (or b_2) to have the faulty value. It is noted that a fault in a_2 cannot propagate to b_1 since b_1 is independent of a_2. However, if a fault changes the value of a_1 to 2 (or changes the value from 2 to either 0 or 1), then the fault also propagates to b_2 since the value of b_2 depends not only on a_2 but also on a_1.

Our approach ensures detection of a single fault in any M-S gate and permutative gate even though the fault may propagate to multiple lines. Proofs are omitted due to page limitations. However, this testable circuit is unable to detect a fault in Feynman gates and modified Feynman gates, which are added to make the circuit testable. Ongoing work is addressing this.

4 Discussion

We have implemented a non-testable ternary half adder from the truth table using the method described in Section 2.5. The cost of the circuit when built in this way is 50 (based on the cost metrics given in [10]). We have applied our approach to convert this into a testable adder with a final cost of 85. The overhead of adding the testability is in this case 70%. We calculate the fault coverage of our circuit to be 84.89% based on the single-bit fault model from [5].

We point out that our approach adds exactly p Feynman gates, $p+q$ modified Feynman gates, $p-1$ M-S gates, and $2p-2$ permutative gates regardless of the number of gates in the given circuit. This results in a higher overhead cost for a small circuit such as ternary half adder. For circuits with higher numbers of gates the overhead costs will be reduced. In addition a larger circuit will cover a higher number of detectable faults since the number of Feynman gates and modified Feynman gates becomes smaller compared to other gates in a large circuit. Future work includes computation of the overhead cost and fault coverage for large benchmark circuits.

5 Conclusion

We have introduced a technique that takes a ternary reversible circuit generated as described in [10] and transforms the circuit into an online testable circuit. This is achieved through the addition of three additional lines and p Feynman gates, $p+q$ modified Feynman gates, $p-1$ M-S gates, and $2p-2$ permutative gates where p is the number of input lines and q is the number of output lines.

Our preliminary work is showing similar overhead to that in [4]; that is, for small circuits the overhead percentage is high, but as gate counts increase this is reduced. The resulting fault coverage seems quite good, although further work on larger benchmarks is continuing. Future work will also include further analysis and comparisons to related work such as [6] and [5].

References

1. Landauer, R.: Irreversibility and heat generation in the computing process. IBM Journal of Research and Development **5** (1961) 183–191
2. Bennett, C.H.: Logical reversibility of computation. IBM Journal of Research and Development **17**(6) (1973) 525–532
3. Nielsen, M., Chuang, I.: Quantum Computation and Quantum Information. Cambridge University Press (2000)
4. Nayeem, N.M., Rice, J.E.: A new approach to online testing of TGFSOP-based ternary Toffoli circuits. In: Proceedings of the 42nd International Symposium on Multiple-Valued Logic (ISMVL), Victoria, BC, Canada (May 14-16 2012) (to appear).
5. Vasudevan, D.P., Lala, P.K., Jia, D., Parkerson, J.P.: Reversible logic design with online testability. IEEE Transactions on Instrumentation and Measurement **55**(2) (2006) 406–414
6. Mahammad, S.N., Veezhinathan, K.: Constructing online testable circuits using reversible logic. IEEE Transactions on Instrumentation and Measurement **59**(1) (2010) 101–109
7. Wang, L., Wu, C., Wen, X., eds.: VLSI Test Principles and Architectures: Design for Testability. Morgan Kaufmann (2006)
8. Hayes, J.P., Polian, I., Becker, B.: Testing for missing-gate faults in reversible circuits. In: Proceedings of the 13th Asian Test Symposium. (2004) 100–105
9. Khan, M.H.A., Perkowski, M., Khan, M.R., Kerntopf, P.: Ternary GFSOP minimization using Kronecker decision diagrams and their synthesis with quantum cascades. Journal of Multiple-Valued Logic and Soft Computing **11**(5-6) (2005) 567–602
10. Khan, M.H.A., Perkowski, M.A.: Quantum ternary parallel adder/subtractor with partially-look-ahead carry. Journal of Systems Architecture **53**(7) (2007) 453–464
11. Rahman, M.R.: Online testing in ternary reversible logic. Master's thesis, University of Lethbridge (2011)
12. Khan, M.H.A., Perkowski, M., Kerntopf, P.: Multi-output Galois field sum of products synthesis with new quantum cascades. In: Proceedings of the 33rd International Symposium on Multiple-Valued Logic. (2003) 146–153
13. Miller, D.M., Maslov, D., Dueck, G.W.: Synthesis of quantum multiple-valued circuits. Journal of Multiple-Valued Logic and Soft Computing **12**(5-6) (2006) 431–450
14. Khan, M.H.A.: GFSOP-based ternary quantum logic synthesis. In: Proceedings of SPIE 7797 (Optics and Photonics for Information Processing IV), San Diego, California (2010)